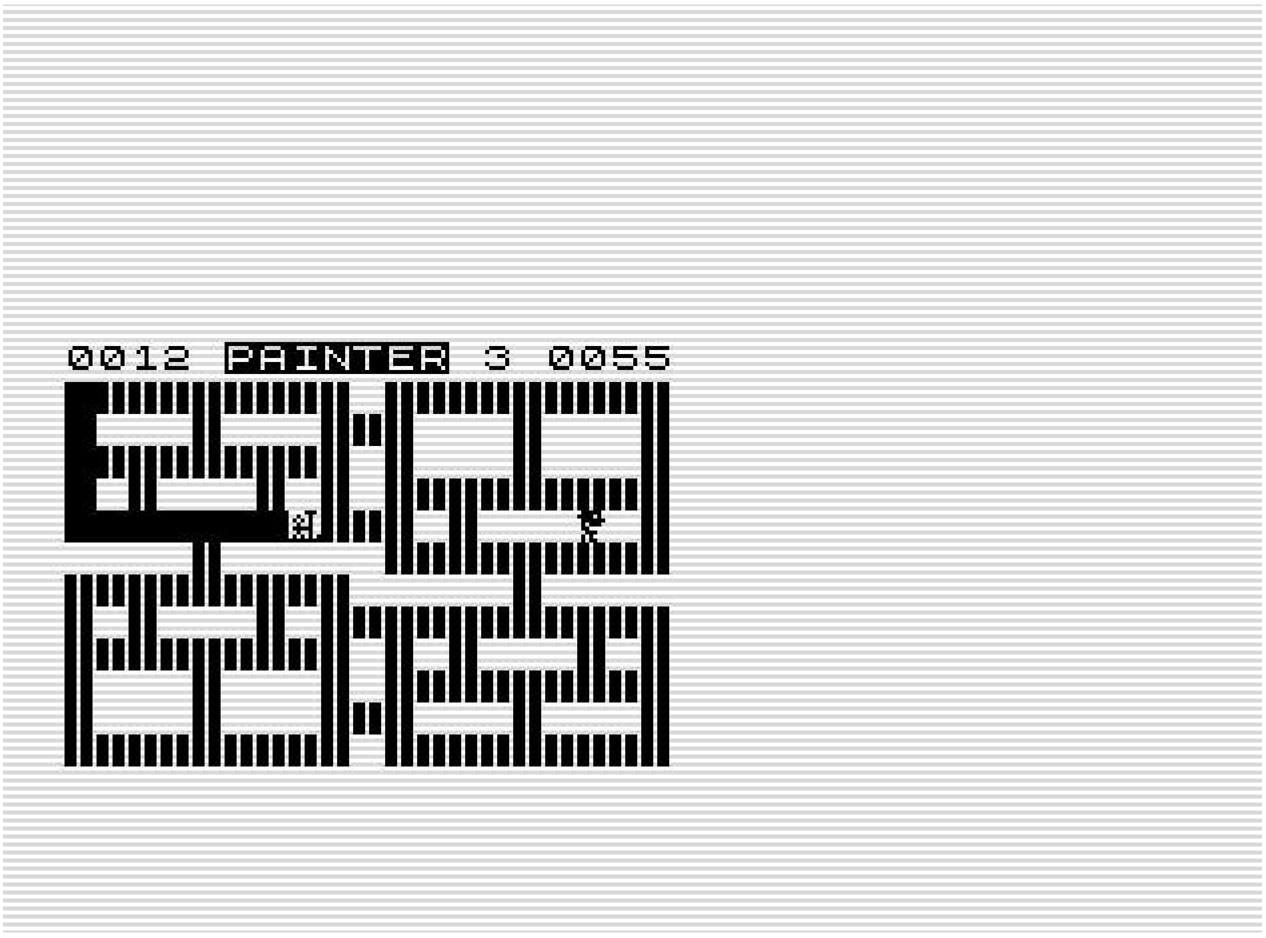
Painter



**I would use the screen from DODGEMS in other games. This is one of them.
The idea is based on 3D PAINTER on the ZX Spectrum (which is NOT 3D, BTW) but with added effects like multiple enemies, multiple paintlayers and speeding up.**

```
; Painter
; Game 55 in 1K hires for the ZX81
; with ZXPAND Joystick support

? * TORNADO *

           ORG  #4009              ;#4009
           DUMP 49161

s3         EQU  init

dirtab     EQU  #400A

basic      LD   D,#C0              ; preset for 48K bug
           JR   init0              ; this game has no 48K bug

           DEFB 236,212,28         ; The BASIC
           DEFB 126                ; fully placed over sysvar
           DEFB 143,0,18           ; start to BASIC=#4009

eline      DEFW last               ; needed to load
chadd      DEFW last-1
xptr       DEFW 0
stkbot     DEFW last               ; needed to load
stkend     DEFW last               ; needed to load
```

```
berg        DEFB 0
mem         DEFW 0
            DEFB 0                ; 128

init1       JP   init             ; init can be anywhere

; all above reusable AFTER loading

lastk       DEFB 255,255,255      ; used by ZX81
margin      DEFB 55               ; used by ZX81
nxtlin      DEFW basic            ; reusable after load

init0       LD   E,L              ; delay intrupts by
            DEFB #16              ; LD D,#40
flagx       DEFB 64               ; clever setting of flags

dirs        XOR  A                ; intruptcounter reset
            EX   AF,AF'

taddr       DEFW 0                ; used by ZX81,no hurting code
            LD   B,0              ; frames is set ok

frames      DEFW #DD*256+1        ; used by ZX81, clever IX set
coprcc      LD   HL,hr            ; set IX
sposn       JR   init1
cdflag      DEFB 64               ; used by zx81

u1          EQU  7
up          EQU  udgplay*256/256

setbg       DEFB 255,85,36,124,36,84,36
udgplay     DEFB 14               ; the player udg

screensp    DEFW u1+#FE00         ; pos udg computer
            DEFW up+#FE00         ; pos udg player
            DEFW s1*256           ; background pointer

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW s2*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW s3*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW s4*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW s5*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW s6*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW s7*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW s8*256
            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW s9*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW sa*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW sb*256

            DEFW u1+#FE00
            DEFW up+#FE00
            DEFW sc*256

            DEFW #FE00
            DEFW #FE00
            DEFW 1                  ; set C for return

hr          LD   HL,lowres+#8000    ; the lowres display
            LD   BC,#241            ; minimum needed
            LD   A,#1E
            LD   I,A
            LD   A,#FB
            CALL #2B5               ; display lowres

            EXX                     ; program uses shadowregs
            PUSH BC                 ; must be saved too
            PUSH DE
            PUSH HL

bgpos       LD   DE,#FEFE
            LD   H,s1/256
            LD   L,E
            LD   C,(HL)             ; get background pos1
            LD   L,D
            LD   B,(HL)             ; get background pos2
            PUSH BC                 ; save backgrounds
            PUSH DE                 ; save pointers

            LD   B,11               ; outline delay for hires
hr00        DJNZ hr00

            LD   (savesp+1),SP      ; save current stack
            LD   SP,screensp        ; use display stack
            LD   A,H
            LD   I,A
            LD   D,A
            LD   H,#40
            EXX
            LD   D,A
            LD   H,#40

bloop       DEFB #DD
            LD   L,7
            POP  BC                 ; get x1 and udg1
            EXX
            POP  BC                 ; get x2 and udg2
            POP  AF                 ; get background pointer+flag

nline       RET  M                  ; filler
```

```
            LD   E,B                ; set x2
            LD   L,C                ; set udg2 pointer
            LDI                     ; write udg to screenline
            EXX
            LD   E,B                ; set x1
            LD   L,C                ; set udg1 pointer
            LDI                     ; write udg to screenline
            JP   NC,#C008           ; do display with LBUF

savesp      LD   SP,0               ; retrieve stack

            POP  DE                 ; get positions
            POP  BC                 ; get backgrounds
            LD   H,s1/256

            LD   L,E
            LD   (HL),C             ; repair background baddie1

            LD   L,D
            LD   (HL),B             ; repair background baddie2

            POP  HL
            POP  DE
            POP  BC
            EXX

            CALL #292               ; generate blank lines bottom
            CALL #220               ; read keyboard, update FRAMES
            LD   IX,hr              ; set HR routine back
            JP   #2A4               ; exit intrupt

cloop       EXX
            DEFB #DD
            DEC  L

            NOP                     ; filler
            LD   E,(HL)             ; filler

            JR   nline

deadtest    PUSH BC
            EXX
            EX   (SP),HL
            AND  A
            SBC  HL,BC              ; XY baddie = XY player?
            POP  HL
            EXX
            RET  NZ

            LD   SP,s1-1            ; repair StackPointer
            LD   HL,lives
            DEC  (HL)               ; 1 live less
            LD   A,(HL)
            CP   28
            JR   NZ,nxtlive

            LD   HL,score-1         ; game over, hiscore test
            LD   DE,hiscore-1
            LD   BC,5
fihi        INC  HL
            INC  DE
            DEC  C
            LD   A,(DE)             ; when C=0 (DE)=118
            CP   (HL)               ; and (HL)=0
```

```
           JR   Z,fihi             ; so NOT equal and no
           CALL C,#19F9            ; hiscore with same score

start      LD   A,(lastk)          ; game over, wait for
           SUB  %10111111          ; newline
           JR   NZ,start

           LD   (entest+1),A       ; set 1 enemy

loadstart  LD   HL,#1C1C           ; reset score
           LD   (score),HL
           LD   (score+2),HL

           LD   A,31               ; set 3 lives
           LD   (lives),A

; first  1 or 2 enemies
; second up to 3 multilayer paint
; last   speed up

           LD   A,248              ; reset speed up
nspeed     LD   (delay+1),A
           INC  A
           JR   Z,nspeed-2

           LD   HL,sd+3            ; reset layers
npaint     LD   (paintnr+1),HL

           LD   HL,#400E
           LD   A,24               ; JR , reset enemies
nenemy     LD   (badscreen),A
           LD   DE,nxtlin          ; store xy enemies
           LD   BC,4
           LDIR

paintnr    LD   HL,0
           LD   E,(HL)             ; read start painting
           LD   HL,sd
           LD   B,228              ; screen has 228 positions
           LD   (HL),148           ; with 148 paintable fields
cls        DEC  HL
           LD   A,(HL)
           OR   A
           JR   Z,skip             ; test paintable
           LD   (HL),E             ; set paintfield
skip       DJNZ cls

           LD   B,4                ; a nice flashing
show0      LD   HL,#401C           ; on every new screen
show1      LD   A,128              ; to paint
           XOR  (HL)
           LD   (HL),A
           DEC  HL
           LD   A,L
           CP   9
           JR   NZ,show1
           CALL delay              ; delay needed to show flash
           DJNZ show0

nxtlive    LD   A,200              ; fixed time
           CALL delay+2            ; extra delay before (re)start

           LD   B,12
clrudg     DEC  B
```

```
           CALL field
           LD   A,254
           LD   (DE),A            ; erase baddie
           INC  DE
           INC  DE
           LD   (DE),A            ; erase player
           INC  B
           DJNZ clrudg

           LD   C,B
ploop      CALL field
           LD   A,254
           INC  DE
           INC  DE
           LD   (DE),A            ; erase display
           PUSH BC

zxpand     LD   BC,%1110000000000111
           LD   A,#A0
           OUT  (C),A
           INC  HL                ; 12 tstates delay
           DEC  HL
           IN   A,(C)
           LD   B,5
           LD   HL,#400A-1
zxp2key    INC  HL
           ADD  A,A
           JR   NC,dirfound       ; a ZXPAND joystick is used
           DJNZ zxp2key

           LD   BC,(lastk)
           LD   A,C
           INC  A
           CALL NZ,#7BD           ; when a key pressed find ASCII
dirfound   LD   A,(HL)            ; read ASCII, both ZXPANS and keyboard

           POP  BC
           CALL checkdir
           CALL field

           CP   238               ; final paintlayer?
           JR   NZ,skipdec

           PUSH HL
           LD   HL,score+4        ; we score a point
           DEFB #3A
ten        LD   (HL),28
           DEC  HL
           INC  (HL)
           LD   A,(HL)
           CP   38
           JR   Z,ten

           LD   HL,sd
           DEC  (HL)              ; 1 field less to paint
           POP  HL

           JR   NZ,skipdec        ; not filled yet

entest     LD   A,0               ; loop add enemie
           LD   HL,#4010
           XOR  H                 ; switch 0/64 and 64/0
           LD   (entest+1),A      ; next enemy level set
           LD   A,62               ; LD A,n
           JP   NZ,nenemy

           LD   HL,(paintnr+1)     ; add a layer to paint
           DEC  HL
           LD   A,L
           CP   sd*256/256         ; 3 layers done?
           JP   NZ,npaint

           LD   A,(delay+1)        ; loop add speed
           INC  A
           JP   nspeed             ; set higher speed

skipdec    PUSH HL
           LD   A,(HL)             ; get current background
           LD   HL,sd+1
fpaint     CP   (HL)               ; find in table
           INC  HL                 ; point to next layer
           JR   NZ,fpaint
           LD   A,(HL)             ; get next layer
           LD   (setbg),A          ; set next layer in UDG

           POP  HL
           LD   (HL),A             ; but also set on screen
           INC  DE
           INC  DE
           LD   A,L
           LD   (DE),A             ; show player

           LD   DE,bgpos+1
           LD   HL,dirs
           PUSH BC
           EXX

;handle both enemies,get xy do move place again
           LD   HL,nxtlin
baddie2    LD   B,(HL)             ; get y pos baddie
           INC  HL
           LD   C,(HL)             ; get x pos baddie
           PUSH HL

           CALL deadtest           ; test player moved to enemy

           CALL field              ; get old position
           JR   Z,sk2

           LD   A,250
           LD   (DE),A             ; erase display

; do move
finddir    LD   HL,(frames)
rseed      LD   DE,0
           ADD  HL,DE
           DEC  HL
           LD   A,H
           AND  #1F
           LD   H,A
           LD   (rseed+1),HL
           LD   A,(HL)
           AND  3
           EXX
           CP   (HL)
           EXX
           JR   Z,finddir          ; not in opposite direction
           PUSH AF                ; save direction
           LD   HL,dirtab
           ADD  A,L
           LD   L,A
           LD   A,(HL)
           CALL checkdir
           POP  HL
           JR   Z,finddir         ; goes move to valid field?

badscreen  JR   baddie1           ; hidden baddie can't move

           EXX
           LD   A,#2E
           SUB  L
           AND  250               ; 0 or -6
           EXX
           ADD  A,B               ; B-0 or B-6
           CP   6
           JR   C,baddie1         ; baddie on valid screen

error      EXX
           LD   (HL),H            ; impossible direction
           EXX
           POP  HL                ; get X and Y pointer
           DEC  HL                ; get 1 back
           JR   baddie2           ; do again

baddie1    LD   A,H
           XOR  1
           EXX
           LD   (HL),A
           EXX

           CALL field             ; get new position

           LD   A,L               ; get screenposition

           EXX
           LD   (DE),A            ; save position
           EXX

           LD   (DE),A            ; write to stack, show enemy

sk2        EXX
           INC  HL
           INC  DE
           EXX

           POP  HL
           LD   (HL),C
           DEC  HL
           LD   (HL),B

           CALL deadtest          ; test enemy moved to player

           INC  HL
           INC  HL
           LD   A,nxtlin*256/256+3
           CP   L
           JR   NC,baddie2        ; "show" 2 baddies

           EXX
           POP  BC
           CALL delay

           JP   ploop

checkdir   LD   HL,dirtab
           PUSH BC
           CP   (HL)              ; up
           INC  HL
           JR   NZ,tdown
           DEC  B
tdown      CP   (HL)              ; down
           INC  HL
           JR   NZ,tleft
           INC  B
tleft      CP   (HL)              ; left
           INC  HL
           JR   NZ,tright
           DEC  C
tright     CP   (HL)              ; right
           JR   NZ,tmove
           INC  C
tmove      CALL field
           POP  HL
           RET  NZ
           LD   B,H               ; illegal move
           LD   C,L
           RET

field      LD   DE,sposn          ; error stack in DE
           LD   HL,sc+20          ; error screen in HL
           LD   A,C
           CP   19
exitfield  SBC  A,A
           RET  NC                ; out of screen
           LD   A,B
           CP   12
           JR   NC,exitfield      ; out of screen

           ADD  A,A               ; now calculate stackaddress
           ADD  A,B
           ADD  A,A
           LD   L,A               ; l = 6*b
           ADD  A,screensp*256/256+1
           LD   E,A               ; DE = COMPUTERPOS, PLAYER +2

           LD   A,L               ; a =  6*b
           ADD  A,L               ; a = 12*b
           ADD  A,L               ; a = 18*b
           ADD  A,B               ; a = 19*b
           ADD  A,C
           ADD  A,#0E
           LD   L,A
           LD   A,(HL)
           OR   A                 ; test move to valid field
           RET

delay      LD   A,0
           LD   HL,frames
           ADD  A,(HL)
wfr        CP   (HL)
           JR   NZ,wfr
           RET

x          EQU  101
```

```
lowres      DEFB 118
score       DEFB 28,28,28,28,0

            DEFB "P"+x,"A"+x,"I"+x,"N"+x,"T"+x,"E"+x
            DEFB "R"+x,0
lives       DEFB 28,0
hiscore     DEFB 28,28,33,33       ; "0055"
            DEFB 118

; stack is set here, but data is moved first
screentab   DEFB 170,186,238,255,255 ; the "colour"-layers

scrcopy     LD   HL,screentab
            LD   DE,sd+1           ; not loadable, but still
            LD   C,5               ; useable memory
            LDIR
            JP   loadstart         ; now autostart the game

space       EQU  #430E-$
            DEFS space

s1          XOR  A                 ; 01 filler
            XOR  A                 ; 02 filler
            LD   HL,s7             ; 05 second part of screen
            LD   B,s7-s1           ; 07 size of second part
            LD   D,H               ; 08
            LD   E,L               ; 09
            NOP                    ; 10
scopy       DEC  HL                ; 11 read first part screen
            LD   A,(HL)            ; 12 backwards
            LD   (DE),A            ; 13 write to second forwards
            INC  DE                ; 14
            DJNZ scopy             ; 16 copy first part
            JP   scrcopy           ; 19 further initialization

;           DEFB 1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1
s2          DEFB 1,0,0,0,1,0,0,0,1,1,1,0,0,0,1,0,0,0,1
;s3          DEFB 1,1,1,1,1,1,1,1,1,0,1,0,0,0,1,0,0,0,1
init        LD   HL,udgc1
            LD   SP,s1
            DEC  SP
            JR   initc

            DEFB 0,1,0,0,0,1,0,0,0,1

s4          DEFB 1,0,1,0,0,0,1,0,1,0,1,1,1,1,1,1,1,1,1
s5          DEFB 1,1,1,1,1,1,1,1,1,1,0,1,0,0,0,1,0,1
s6          DEFB 0,0,0,0,1,0,0,0,0,0,1,1,1,1,1,1,1,1,1
s7          EQU  $
s8          EQU  s7+19
s9          EQU  s8+19
sa          EQU  s9+19
sb          EQU  sa+19
sc          EQU  sb+19
sd          EQU  sc+19

initc       LD   E,B
            LD   C,35
            LDIR

            LD   HL,#4000
            LD   DE,#C000
            LD   C,36
```

```
           LDIR                   ; repair 48K bug
           JP   s1

udgc1      DEFB 88,80,56,96,92,126,244,124

lbuf       LD   R,A               ; 4008  get displayline
           DEFB 54,38,52,53       ; 400a  QAOP directiontable
           DEFB 20,20,0,15,11,0   ; 400e  starttable enemies
           DEFB 0,0,0,0           ; 4014
           DEFB 0,0,0,0           ; 4018
           DEFB 0                 ; 4019
           JP   Z,bloop           ; 4020  48K bug
           JP   cloop             ; 4023  48K bug

vars       DEFB 128
?
last       EQU  $
```